APPLICATION

FOR

UNITED STATES LETTERS PATENT

TITLE: GENERATING A SHADOW FOR A THREE-DIMENSIONAL MODEL

APPLICANT: ADAM T. LAKE AND CARL S. MARSHALL
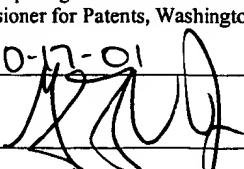
GENERATING A SHADOW FOR A
THREE-DIMENSIONAL MODEL

TECHNICAL FIELD

5   This invention relates to generating a shadow for a
three-dimensional (3D) model based on bones in the
infrastructure of the 3D model.

BACKGROUND

10   A 3D model includes a virtual skeleton/infrastructure
comprised of bones that are arranged in a hierarchical tree
structure. Surrounding the bones is a polygon mesh, comprised
of polygons such as triangles, which represents the skin of
the 3D model. Movement of the polygon mesh is tied to the
15 movement of the bones so that the 3D model approximates real-
life movement when the bones are re-positioned.

     The 3D model inhabits a virtual world, in which the
distance to a virtual camera dictates perspective. A virtual
light source, positioned in the environment of the virtual
20 world, is used as a reference point for projecting shadows of
the 3D model onto surfaces in the environment.

DESCRIPTION OF THE DRAWINGS

Fig. 1 is a view of a 3D model.

Fig. 2 is a view of bones in the 3D model of Fig. 1.

Fig. 3 is a flowchart showing a process for generating a

5   shadow for the 3D model.

Fig. 4 is a view of a 3D model polygon mesh.

Fig. 5 is a view of bones in the 3D model of Fig. 4.

Fig. 6 is a perspective view of a 3D environment.

Fig. 7 is a block diagram showing projection of a bone

10   onto a surface.

Fig. 8 is a top view of a bone projection showing

creation of a shape around the projection.

Fig. 9 is a view of a 3D model of a tree.

Fig. 10 is a top view of shadows of leaves in the tree.

15   Fig. 11 is a flowchart showing another process for

generating a shadow for a 3D model using bone bounding

volumes.

Fig. 12 is a block diagram showing projecting a bounding

volume of a bone onto a surface.

20   Fig. 13 is a block diagram of a computer system on which

the processes of Figs. 3 and/or 11 may be executed.

DESCRIPTION

Fig. 1 shows a 3D model 10, which is rendered from 3D data. 3D model 10 is comprised of a polygon mesh (not shown). The polygons are triangles in this embodiment; however, other types of polygons may be used. The polygon mesh defines the "skin" surface of 3D model 10.

The 3D data for model 10 also includes bone data. The bone data defines a rigid infrastructure 12 of model 10 (Fig. 2). The infrastructure corresponds to the skeletal structure of a living being. In this embodiment, the "bones" that make up the infrastructure are Cartesian XYZ-space vectors.

Fig. 3 shows a process 14 for generating shadows of 3D model 10 using its bones. Process 14 includes two phases: a pre-processing phase 16 and a run-time phase 18. Pre-processing phase 16 need not be repeated for each frame of 3D animation. That is, pre-processing phase 16 may be performed only once at the beginning of an animation sequence or, alternatively, it may be repeated as desired. Run-time phase 18, generally speaking, is repeated for each frame of animation, although this is not necessarily a requirement.

In pre-processing phase 16, process 14 receives (301) data that corresponds to the size and/or shape of a shadow to

be generated. The data may be input by an animator (user) via a graphical user interface (GUI) (not shown) or the like.

Process 14 reads (302) 3D data that defines the geometry of a frame of 3D animation. The 3D data may be read from a local memory, from a remote database via a network such as the Internet, or obtained from any other source. The data includes 3D data that defines a 3D model, including its polygon mesh (see, e.g., Fig. 4 for 3D model 20) and its bone structure (see, e.g., Fig. 5 for 3D model 20).

Process 14 locates (303) a virtual camera and virtual light source in the environment of the read frame of 3D animation. The location of a virtual camera defines a viewer's perspective for the frame of 3D animation. For example, in Fig. 6, the virtual camera is located at plane 22, making object 24 appear closer in 3D environment 26 than object 28. A virtual light source 30 is positioned within 3D environment 26 and defines how light hits objects in that environment. For example, as shown Fig. 6, a vector 32 corresponds to a ray of light from virtual light source 30, which affects the illumination of object 24.

In Fig. 3, during run-time phase 18, process 14 identifies the bones infrastructure of a 3D model (e.g., 3D model 10 or 20) and, for each bone in the 3D model, proceeds

as follows. Process 14 projects (304) the bone onto a surface

in the 3D environment. Fig. 7 shows a bone 34 being projected

onto a surface 36 in a 3D environment. In more detail, to

project bone 34, process 14 draws (304a) lines (e.g., vectors)

5   from virtual light source 38 through end points 40 and 42 of

bone 34. These lines extend to surface 36, namely, to points

44 and 46 on surface 36, as shown in Fig. 7. Process 14

connects (304b) points 44 and 46 on surface 36, resulting in a

projection 48 of bone 34 onto surface 36.

10   Process 14 generates (305) a shadow on surface 36 based

on projection 48 of bone 34. Generating a shadow for each

bone in a 3D model results in a shadow for the entire 3D model

itself. It is noted that process 14 may not generate a shadow

for every bone in a 3D model. For example, process 14 may

15   generate shadows for only "major" bones in a 3D model, where

major bones may be defined, e.g., by their length (bones that

are greater than a predefined length) or proximity to the

trunk/body of a 3D model (bones that are within a predefined

number of bones from a predefined reference bone).

20   Process 14 generates the shadow based on the data

received (301) in pre-processing phase 16. That is, the data

defines the size and shape of the shadow. Process 14

therefore generates the shadow accordingly. This is done by

creating (305a) a shape over at least part of projection 48 of
the bone. The shape may be created, e.g., by growing a
polygon from projection 48 (for the purposes of this
application, the definition of "polygon" includes smooth-edged
5   shapes, such as a circle, ellipse, etc.).

By way of example, referring to Fig. 8, process 14 may
grow a quadrilateral 50 over projection 48 of bone 34. More
specifically, using projection 48 as the medial axis, process
14 projects vectors 52 outward from projection 48. The
10   vectors, when connected, create a polygon that bounds the
projection. The size and shape of that polygon are dictated
by the size and shape of the projection and the data received
during pre-processing phase 16. For example, if a larger
shadow is desired, vectors 52 will be longer, resulting in a
15   larger shape than if a smaller shadow is desired.

Process 14 maps (305b) one or more textures onto the
shape (e.g., quadrilateral 50) that was created over
projection 48. The texture(s) may define a color of the shape
as well as how transparent or translucent the shadow is. That
20   is, it may be desirable to see objects covered by the shadow.
Therefore, a light color that is relatively transparent may be
mapped. For example, texture with an alpha transparency value
of 50% may be used for the mapping.

A "fuzzy" texture may also be mapped onto edges or other portions of the shape. In this context, a fuzzy texture is a texture that does not have sharp edges, meaning that the edges fade out from darker to lighter (hence the use of the term

5   "fuzzy"). Fuzzy textures provide softer-looking shadows, which can be difficult to construct using other techniques.

It is noted that process 14 may be used with other animation that does not necessarily have a bones-based infrastructure. In this case, bones may be defined for a 3D

10   model and then process 14 may be applied. For example, bones may be defined for the veins of leaves on a tree 56 (Fig. 9). Process 14 may project shadows 58 (Fig. 10) of the leaves onto ground 59 from, e.g., a virtual sun 60 (light source).

As another example, process 14 may be used to generate a

15   shadow of a ball (not shown). In this example, a spherical "bone" or a linear bone that points to the virtual light source (i.e., that looks like a point relative to the virtual light source) may be used to represent the ball. The bone may be projected onto a surface and a shape, such as a circle or

20   an ellipse, may be grown from the projection. The type of shape that is grown may be defined by the user-input data or it may be determined by analyzing the shape of the bone. For

example, a spherical bone may dictate a circular shape and a linear bone may dictate a rectangular shape.

Referring to Fig. 11, an alternative process 61 is shown for generating a shadow of a 3D model having a bones-based
5   infrastructure. Process 61 includes a run-time phase 64 and a pre-processing phase 62 that is identical to pre-processing phase 16 of process 14. Accordingly, a description of pre-processing phase 62 for process 61 is not repeated here.

During run-time phase 64, process 61 identifies the bones
10   infrastructure of a 3D model (e.g., 3D model 10 or 20) and, for each bone in the 3D model, proceeds as follows.

Process 61 generates (1101) a bounding volume for the bone. The bounding volume of the bone is an expansion of a two-dimensional (2D) bone into 3D space. Referring to Fig.
15   12, the bounding volume 65 of bone 66 may be generated by projecting a vector 68 from bone 66 and rotating the vector around the bone in the direction of arrow 70. For example, the bounding volume of a vector may be a cylinder, the bounding volume of a point is a sphere, etc. The size and
20   shape of the bounding volume may be dictated by the data received from the user. For example, if a large shadow is desired, a large bounding volume is generated.

Process 61 generates (1102) a shadow of bone 66 by projecting (1102a) a shape of bounding volume 65 onto surface 72. In more detail, process 61 draws lines (e.g., vectors) 73 from virtual light source 74, through locations on the surface (e.g., the perimeter) of bounding volume 65, onto surface 72. The number of lines drawn depends on the shape of bounding volume 65. For example, if bounding volume 65 is a cylinder (as shown), a number of lines (e.g., four) may not be required to project the shadow. On the other hand, if bounding volume 65 is a sphere, more bounding lines may be required to achieve a shadow that is a relatively close approximation of the shape of the bounding volume.

To project the shape, process 61 connects points, in this example, points 77, 78, 79 and 80 at which lines 73 intersect surface 72. Connecting the points results in a shape 83 that roughly corresponds to the outline of bounding volume 65 relative to virtual light source 74.

Process 61 maps (1102b) one or more textures onto shape 83 created by connecting the points. The texture mapping, and contingencies associated therewith, are identical to the texture mapping described above with respect to process 14.

As was the case above, process 61 may not generate a bounding volume and shadow for every bone in a 3D model. For

example, process 61 may generate shadows for only "major"

bones in a 3D model, where major bones may be defined, e.g.,

by their length or proximity to the trunk/body of a 3D model.

Fig. 13 shows a computer 82 for generating shadows of 3D

5   models using processes 14 or 61. Computer 82 includes a

processor 84, a memory 88, and a storage medium 90 (e.g., a

hard disk) (see view 92). Storage medium 90 stores 3D data

94, which defines a 3D model, and machine-executable

instructions 96 and 98, which are executed by processor 84 out

10   of memory 88 to perform processes 14 and/or 61 on 3D data 94.

Processes 14 and 61 both have the advantage of using

existing data, e.g., bones, to generate a shadow using

relatively little computational effort. Moreover, processes

14 and 61 also give the user control over the look and feel of

15   the resulting shadow.

Processes 14 and 61, however, are not limited to use with

the hardware and software of Fig. 13; they may find

applicability in any computing or processing environment.

Processes 14 and 61 may be implemented in hardware,

20   software, or a combination of the two. Processes 14 and 61

may be implemented in computer programs executing on

programmable machines that each includes a processor, a

storage medium readable by the processor (including volatile

and non-volatile memory and/or storage elements), at least one input device, and one or more output devices. Program code may be applied to data entered using an input device, such as a mouse or a keyboard, to perform processes 14 and 61 and to

5   generate output information.

Each such program may be implemented in a high level procedural or object-oriented programming language to communicate with a computer system. However, the programs can be implemented in assembly or machine language. The language

10   may be a compiled or an interpreted language.

Each computer program may be stored on a storage medium or device (e.g., CD-ROM, hard disk, or magnetic diskette) that is readable by a general or special purpose programmable computer for configuring and operating the computer when the

15   storage medium or device is read by the computer to perform processes 14 and 61. Processes 14 and 61 may be implemented as articles of manufacture, such as a machine-readable storage medium, configured with a computer program, where, upon execution, instructions in the computer program cause the

20   machine to operate in accordance with processes 14 and 61.

The invention is not limited to the embodiments described above. For example, elements of processes 14 and 61 may be combined to form a new process not specifically described

herein. The blocks shown in Figs. 3 and 11 may be reordered to achieve the same results as described above.

Other embodiments not described herein are also within the scope of the following claims.

5   What is claimed is: